



**UNIVERSIDAD TECNOLÓGICA
NACIONAL
FACULTAD REGIONAL
RESISTENCIA**

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

MATERIA: SISTEMAS DE GESTIÓN DE BASES DE DATOS

NIVEL: 3

CICLO LECTIVO: 2023 – 2DO. CUATRIMESTRE

DOCENTES:

PROFESOR: I.S.I. ANDRÉS PABLO FANTÍN

J.T.P.: I.S.I. JUAN CARLOS FERNÁNDEZ

CARPETA DE TRABAJOS PRÁCTICOS

Trabajo Práctico N°2: Diseño físico



- Todo el proceso de creación de objetos de base de datos, salvo en los ejercicios en que se especifique lo contrario, debe hacerse obligatoriamente **con comandos escritos** (en otras palabras, NO DEBEN USARSE LOS ASISTENTES GRÁFICOS)
- Crear las tablas **utilizando la mayor cantidad de parámetros posibles** de acuerdo a la documentación de referencia, lo visto en la cátedra y justificando la utilización de cada uno de ellos.
- La **implementación de las restricciones** puede hacerse ya sea desde la creación de la tabla (CREATE TABLE) o luego de creada la misma (ALTER TABLE).
- Deberá **completar** cada punto que requiera la ejecución de sentencias SQL **con el texto de las mismas y una captura de pantalla de la ejecución y el resultado** (de la terminal de comandos del SO o una pestaña de consultas de MySQL Workbench). Ejemplos:

Consigna: Crear una base de datos con el año actual como nombre y dentro de la misma una tabla con distintos campos.

```
CREATE DATABASE bda2019;
USE bda2019;
CREATE TABLE tabla1(
    id int,
    campo1 VARCHAR(20),
    PRIMARY KEY (id)
);
```

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> create database bda2019;
Query OK, 1 row affected (0,03 sec)
```

```
mysql> use bda2019;
Database changed
```

```
mysql> create table tabla1(
  -> id int,
  -> campo1 varchar(20),
  -> primary key (id)
  -> ;
```

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 4

```
mysql> create table tabla1( id int, campo1 varchar(20), primary key (id));
Query OK, 0 rows affected (0,09 sec)
```

```
mysql> █
```

Local instance 3306

File Edit View Query Database Server Tools Scripting Help

Administration Schemas Query 1

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Query 1

```
1. create database bda2019;
2. use bda2019;
3. create table tabla1(
4.     id int,
5.     campo1 varchar(20),
6.     primary key (id)
7. );
```

Action Output

#	Time	Action	Message	Duration / Fetch
1	00:38:25	create database bda2019	Success	0,00098 sec
2	00:38:25	create table tabla1(id int, campo1 varchar(20), pri...	Error Code: 1046. No database selected Select the default DB to be used by double-clickin...	0,00098 sec
3	00:38:44	create database bda2019	1 row(s) affected	0,027 sec
4	00:38:44	use bda2019	0 row(s) affected	0,0038 sec
5	00:38:44	create table tabla1(id int, campo1 varchar(20), pri...	0 row(s) affected	0,079 sec

Object Info Session

No object selected

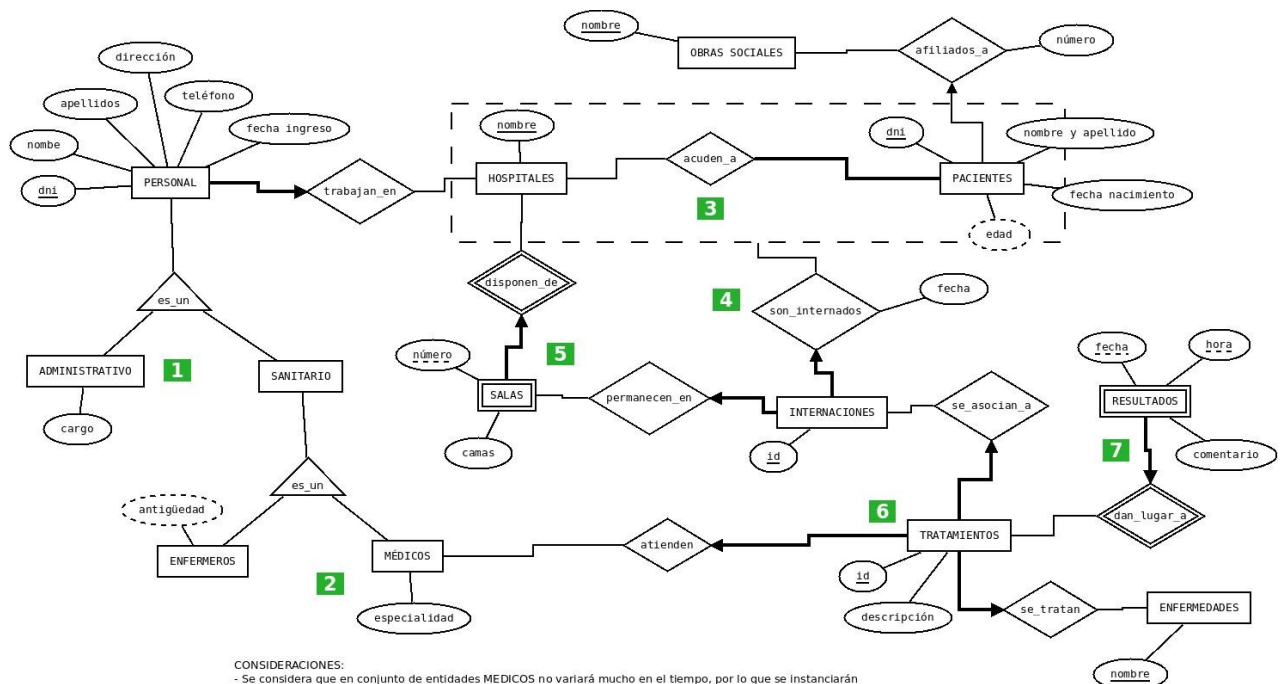
Query Completed

Ejercicio 1: Consorcio de hospitales

Dado el siguiente enunciado y DER resultante:

Se trata de diseñar la base de datos para la administración de un consorcio de hospitales, que permita gestionar datos acerca del personal así como de los pacientes de los mismos. De cada hospital interesa almacenar, además de su nombre dirección, teléfono, fax, etc., lo siguiente:

1. El personal de los hospitales (del que interesa almacenar su dni, nombre, apellido, dirección, teléfono y fecha de ingreso a trabajar al hospital) se divide en personal administrativo (se registra su cargo) y personal sanitario, dentro del que se distingue a su vez enfermeros (se registra su antigüedad en años) y médicos.
2. Los médicos tienen una especialidad que interesa conocer (pediatría, obstetricia, etc.) y sólo trabajan, al igual que el resto del personal, en un hospital.
3. Los pacientes pueden acudir a varios hospitales del consorcio.
4. Se desea conocer los datos personales de los pacientes que van a ingresar en el hospital, así como su obra social, número de afiliado, la fecha de admisión y la sala (habitación) en la que deben permanecer en caso de ser internados.
5. Cada sala se identifica por un número de sala dentro de cada hospital y se desea conocer el número de camas de las que dispone cada sala.
6. Cada admisión de un paciente en el hospital lleva asociada una o varias fichas de tratamiento en las que se indica la enfermedad y el médico que la atiende.
7. Cada tratamiento da lugar a distintos resultados que permiten realizar el seguimiento de cada enfermedad de un paciente. El resultado debe indicar la fecha y hora en que éste tuvo lugar, así como un comentario (por ejemplo, indicando si el paciente tiene fiebre etc.). Para un mismo tratamiento sólo puede haber un resultado en un mismo día, a una misma hora.



CONSIDERACIONES:

- Se considera que en conjunto de entidades MEDICOS no variará mucho en el tiempo, por lo que se instanciarán una sola vez y en ese caso será más adecuado plantear la especialidad como atributo en vez de entidad.
- Se considera paciente a una persona que desde la primera vez que acude a un hospital por cualquier tipo de consulta.
- De la atención de un paciente en un hospital puede surgir la necesidad de internarlo.
- Se debe generar un atributo "id" para identificar los tratamientos y las internaciones.

1. Describir el proceso realizado para pasar al esquema relacional de la base de datos correspondiente, detallando decisiones de diseño tomadas durante el mismo. Si hay elementos que aún no sabe cómo o no tiene herramientas para implementarlos (p.e. algunas restricciones), escríbalos.
2. Realizar un diccionario de datos que contenga información para cada tabla de la siguiente manera:

Nombre de tabla	Descripción / comentarios			
Nombre de columna	Tipo de columna	Atributos/opciones (PK, NOT NULL, AUTO INCREMENTAL, UNIQUE, ETC.)	Valor predeterminado	Descripción / comentarios

3. Escribir el script SQL con las sentencias de creación de los objetos en el motor de base de datos y mostrar los resultados de su ejecución.

Ejercicio 2: Cursos de formación

Dado el siguiente enunciado y DER resultante:

El departamento de formación de una empresa desea construir una base de datos para planificar y gestionar la formación de sus empleados.

La empresa organiza cursos internos de formación de los que se desea conocer el código de curso, el nombre, una descripción, el número de horas de duración y el coste del curso.

Un curso puede tener como prerequisite haber realizado otro(s) previamente, y a su vez la realización de un curso puede ser prerequisite de otros. Un curso que es un prerequisite de otro puede serlo de forma obligatoria o sólo recomendable.

Un mismo curso tiene diferentes ediciones, es decir, se imparte en diferentes lugares, fechas y con diferentes horarios (intensivo, de mañana o de tarde). En una misma fecha de inicio sólo puede impartirse una edición de un curso.

Los cursos se imparten por personal de la propia empresa.

De los empleados se desea almacenar su código de empleado, nombre y apellidos, dirección, teléfono, NIF, fecha de nacimiento, nacionalidad, sexo, firma y salario, así como si está o no capacitado para impartir cursos.

Un mismo empleado puede ser docente en una edición de un curso y alumno en otra edición, pero nunca puede ser ambas cosas a la vez (en una misma edición de curso o lo imparte o lo recibe).

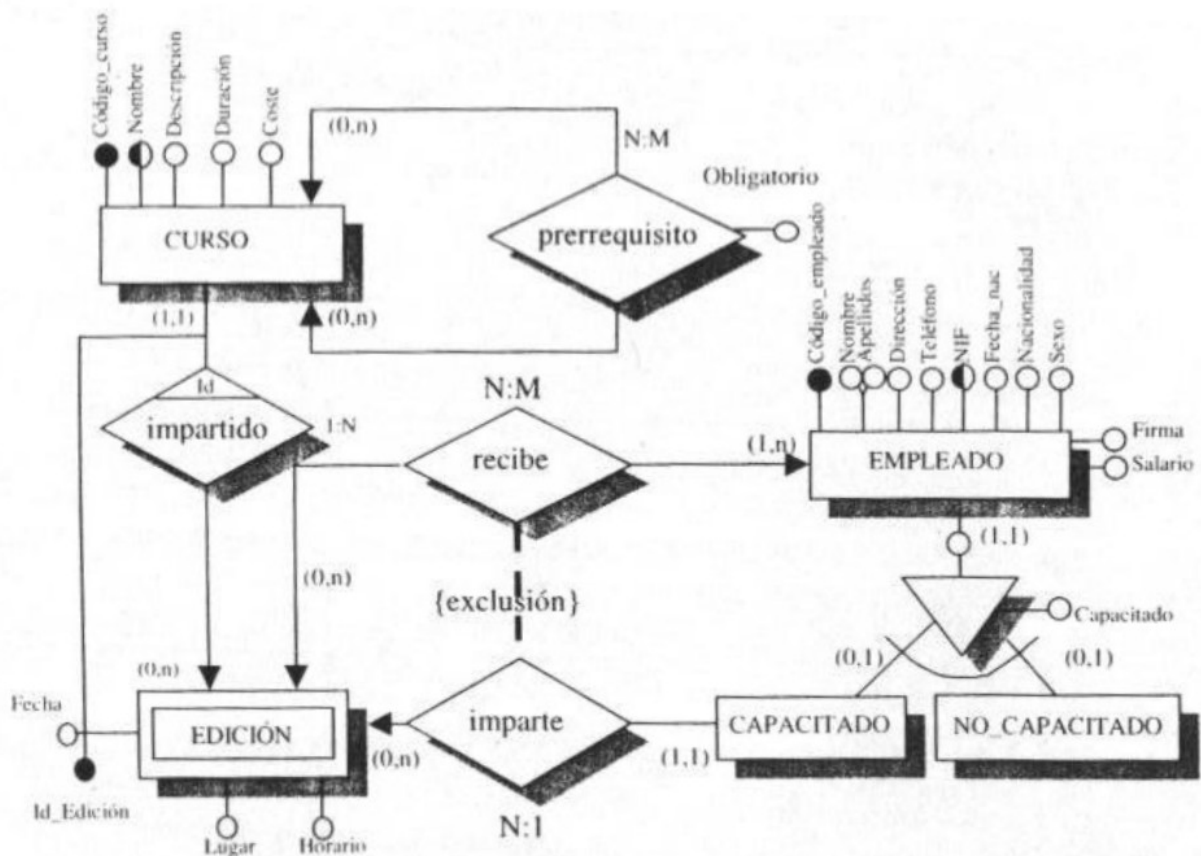


Figura 1.13

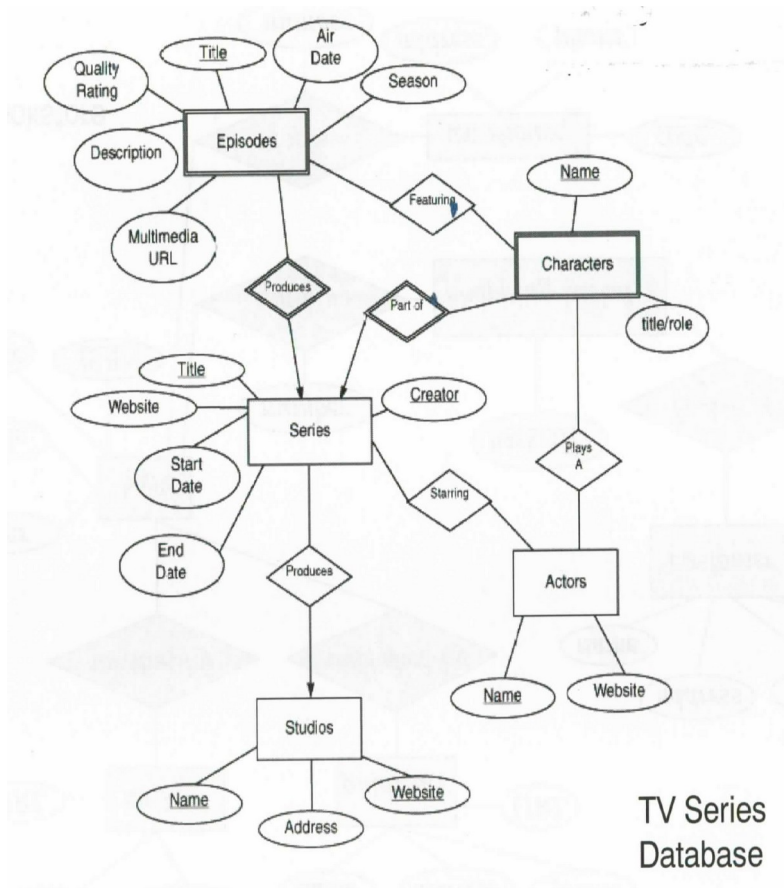
SUPUESTOS SEMÁNTICOS COMPLEMENTARIOS Y SEMÁNTICA NO REFLEJADA

- Se ha considerado el nombre del curso como un identificador alternativo.

Realizar los mismos pasos que en el ejercicio anterior.

Ejercicio 3: Series de TV

Dado el siguiente DER convertir el mismo a tablas y definir los scripts de creación de los objetos en el motor de base de datos.



Ejercicio 4: Alquiler de películas

Se presentan a continuación la descripción y los requerimientos de un caso de estudio, con su respectivo modelo de datos:

La empresa Sakila tiene como objetivo principal el alquiler de películas de videos, manejando una gran variedad de películas con el fin de satisfacer todas las necesidades de sus clientes.

La empresa Sakila tiene varias sucursales, donde sus clientes podrán encontrar las mejores películas del mercado distribuidas por idioma, actor, título, categoría, de la mejor manera posible para su fácil acceso dentro de sus tiendas.

El modelo de negocio de Sakila se compone de los siguientes procesos según el análisis de la base de datos:

- **Datos Clientes:** Donde se almacenan los datos de los clientes que por lo menos una vez, han utilizado los servicios ofrecidos en las sucursales de Sakila.
- **Inventario:** Donde se almacena todos los datos sobre los productos (películas) que se tienen en las sucursales, para la administración y el buen funcionamiento de las tiendas.
- **Proceso del Negocio:** Aquí se encuentran los datos de los alquileres y pagos, como de las sucursales y el personal.

Usando la base de datos creada en el trabajo práctico nro. 1 (BDA2023) y con el escenario anterior y el modelo de datos disponible en el archivo **sakila.pdf** cree las tablas de la base de datos “en castellano”, es decir con los nombres de tablas y campos en nuestro idioma. Realice las siguientes tareas utilizando los criterios dados:

- 1.1. Crear la tabla **cliente** (customer) y definir su clave principal en la misma instrucción de creación. Agregar un campo “fecha_nacimiento” de tipo fecha. Asegurarse de que cuando se inserte un nuevo cliente, éste sea mayor de edad (edad mayor o igual a 18) controlando la diferencia entre la fecha actual y la de nacimiento. Agregar una columna virtual “edad” como un campo generado (“generated column”).
- 1.2. Continuar con las tablas **empleado** (staff), **alquiler** (rental), **direccion** (address) y **pais** (country).
- 1.3. Crear la tabla **pelicula_actor** (film_actor) con su clave principal (hacer restricción en caso de tener una CP compuesta).
- 1.4. Crear la tabla **pago** (payment) con su clave principal. Hacer las correspondientes restricciones.
- 1.5. Crear la tabla **almacen** (store) con sus campos propios y los referenciados. Sin generar claves.
- 1.6. Completar el ejercicio anterior, con la creación de las claves correspondientes. Añadir a la tabla de **empleado** la columna *sueldo_hora*.
- 1.7. Hacer que no puedan haber dos clientes con el mismo nombre y apellido.
- 1.8. Crear la tabla **ciudad** (city) con su correspondiente CP. Agregar clave foránea ciudad_id (city_id) a la tabla **direccion**. Eliminar la columna de direcciones de la tabla **cliente**.
- 1.9. Agregar columnas *calle_cliente* y *altura_cliente* a la tabla **cliente**. Agregar *domicilio_empleado* (address_id) en tabla **empleado**.
- 1.10. Eliminar tabla **pais**.
- 1.11. Eliminar la columna *email* de la tabla **empleado**. Agregar una tabla **email_empleado** (sin clave primaria).
- 1.12. Establecer clave primaria para **email_empleado**, teniendo en cuenta que ésta correspondería a una entidad débil de **empleado**.
- 1.13. Establecer que los nombres y apellidos de los empleados no tengan valores nulos.
- 1.14. Establecer que no se repita la calle y la altura de los clientes.
- 1.15. Crear las siguientes vistas:
 - 1.15.1. “info_actores”: con los datos de los actores y nombres de las películas en las que actuaron.
 - 1.15.2. “lista_clientes”: con información básica de los clientes incluídas sus direcciones.
 - 1.15.3. “info_peliculas”: con los datos de las películas y los actores que actuaron en ellas.

Ejercicio 5: Índices

- 7.1) Utilizando el ejercicio 5 del TP 2 (Alquiler de películas) realice las siguientes tareas:
 - 7.1.1) Crear un índice para la tabla **alquiler** sobre el campo *fecha_alquiler* (rental_date).
 - 7.1.2) Eliminar el índice creado en el punto anterior.
 - 7.1.3) Crear una columna *localizacion* en la tabla **direccion** que permita registrar coordenadas geográficas. Crear un índice que optimice la búsqueda y ordenamiento de la tabla para el campo agregado.
 - 7.1.4) Crear un nuevo índice sobre la tabla **empleado** sobre los campos Apellido y Nombre
 - 7.1.5) Crear un índice que optimice la búsqueda por ciudad y direccion en la tabla **direccion** (índice compuesto) teniendo en cuenta que se realizan inserciones con mucha asiduidad sobre esta tabla.
- 7.2) Analice y discuta con su grupo la factibilidad para la creación de índices de acuerdo a la documentación del motor con el que está trabajando a efectos de optimizar distintas **consultas**

definidas por el grupo a las bases de datos de los ejercicios 1 a 3. Luego impleméntelos en las bases de datos correspondientes. Documente lo anterior haciendo copias de pantalla y guardando scripts de las instrucciones que crean los índices. Justifique.

Ejercicio 5: Plan de ejecución

1. Instale la base de datos original de Sakila (esquemas y datos) de acuerdo a las instrucciones de instalación (<https://dev.mysql.com/doc/sakila/en/sakila-installation.html>)
2. Ejecute y describa qué información arroja la siguiente consulta:

```
SELECT f.title, c.first_name, c.last_name, a.address, ci.city, co.country
FROM film f, inventory i, rental r, customer c, address a, city ci, country co
WHERE f.film_id = i.film_id and i.inventory_id = r.inventory_id
      and r.customer_id = c.customer_id and c.address_id = a.address_id
      and a.city_id = ci.city_id and ci.country_id = co.country_id
      and f.title = 'ACADEMY DINOSAUR' and co.country LIKE 'B%'
ORDER BY co.country, ci.city, c.last_name, c.first_name;
```
3. Ejecute la misma consulta con EXPLAIN y EXPLAIN ANALYSE y tome nota de la información mostrada. De manera similar, ejecute la consulta con MySQL Workbench, haga una captura de pantalla de la pestaña “Execution Plan” y tome nota de la información mostrada.
4. En la tabla de países (Country) cree un índice por el nombre del país (country).
5. Realice nuevamente el punto 3 y haga un cuadro comparativo con los resultados (ejecución de la consulta antes y después de la creación del índice). ¿Qué conclusiones obtiene?

Referencias

- Capítulo 13.1 “Data Definition Statements” del Manual de Referencia de MySQL: <https://dev.mysql.com/doc/refman/8.0/en/sql-data-definition-statements.html>
- Capítulo 8.3.1 “How MySQL Uses Indexes” del Manual de Referencia de MySQL: <https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html>
- Base de datos de ejemplo Sakila: <https://dev.mysql.com/doc/sakila/en/>
- Entendiendo el Plan de Ejecución de Consultas: <https://dev.mysql.com/doc/refman/8.0/en/execution-plan-information.html>
- Sentencia Explain: <https://dev.mysql.com/doc/refman/8.0/en/explain.html>
- Documentación de MySQL Workbench: <https://dev.mysql.com/doc/workbench/en/>
- Foro de la Comunidad MySQL: <http://forums.mysql.com>
- Database Management Systems (ISBN-10: 0072465638 | ISBN-13: 978-0072465631) de Raghu Ramakrishnan, U. de Wisconsin, EE.UU, tercera edición
- Presentaciones de teoría.